IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In Re Application of:		
Applicants	:	Reuven Bakalash et al.
U.S. Application No.	:	10/579,682
Filing Date	:	November 19, 2004
Title of Invention	:	PC-BASED COMPUTING SYSTEM EMPLOYING PARALLELIZED GRAPHICS PROCESSING UNITS (GPUS) INTERFACED WITH THE CENTRAL PROCESSING UNIT (CPU) USING A PC BUS AND A HARDWARE GRAPHICS HUB HAVING A ROUTER
Group Art No.	:	2628
Examiner	:	Hau H. Nguyen
Attorney Docket No.	:	142-002USAC00

Honorable Commissioner of Patents
and Trademarks
Washington, DC 20231

SUPPLEMENTAL INFORMATION DISCLOSURE STATEMENT UNDER 37 C.F.R. 1.97

Sir:

In order to fulfill Applicants' continuing obligation of candor and good faith as set forth in 37 C.F.R. 1.56, Applicants submit herewith a Supplemental Information Disclosure Statement prepared in accordance with 37 C.F.R Sections 1.97, 1.98 and 1.99.

The disclosures enclosed herewith are as follows:

U.S. PUBLICATIONS

NUMBER	FILING DATE	TITLE
7,477,256 B1	November 17, 2004	CONNECTING GRAPHICS ADAPTERS FOR SCALABLE PERFORMANCE
7,372,465	December 17, 2004	SCALABLE GRAPHICS PROCESSING FOR REMOTE DISPLAY
7,325,086 B2	December 15, 2005	METHOD AND SYSTEM FOR MULTIPLE GPU SUPPORT
7,324,547 B1	December 13, 2002	INTERNET PROTOCOL (IP) ROUTER RESIDING IN A PROCESSOR CHIPSET
7,324,111 B2	June 28, 2004	METHOD AND APPARATUS FOR ROUTING GRAPHICS PROCESSING SIGNALS TO A STAND-ALONE MODULE
7,119,808	July 15, 2003	MULTIPLE PARALLEL PROCESSOR

		COMPUTER GRAPHICS SYSTEM
7,051,139	March 31, 2003	CPU EXPANDABILITY BUS
6,885,376	December 30, 2002	SYSTEM, METHOD, AND COMPUTER PROGRAM PRODUCT FOR NEAR-REAL TIME LOAD BALANCING ACROSS MULTIPLE RENDERING PIPELINES
6,864,984	March 14, 2001	MEASURING METHOD AND APPARATUS USING ATTENUATION IN TOTAL REFLECTION
6,844,879	July 16, 2002	DRAWING APPARATUS
6,842,180	September 20, 2000	OPPORTUNISTIC SHARING OF GRAPHICS RESOURCES TO ENHANCE CPU PERFORMANCE IN AN INTEGRATED MICROPROCESSOR
6,801,202 B2	June 28, 2001	GRAPHICS SYSTEM CONFIGURED TO PARALLEL-PROCESS GRAPHICS DATA USING MULTIPLE PIPELINES
6,789,154	May 26, 2000	APPARATUS AND METHOD FOR TRANSMITTING DATA
6,778,177	February 19, 2002	METHOD FOR RASTERIZING A GRAPHICS BASIC COMPONENT
6,753,878	March 8, 1999	PARALLEL PIPELINED MERGE ENGINES
6,741,243	February 8, 2001	METHOD AND SYSTEM FOR REDUCING OVERFLOWS IN A COMPUTER GRAPHICS SYSTEM
6,683,614	December 21, 2001	SYSTEM AND METHOD FOR AUTOMATICALLY CONFIGURING GRAPHICS PIPELINES BY TRACKING A REGION OF INTEREST IN A COMPUTER GRAPHICAL DISPLAY SYSTEM
6,578,068	August 31, 1999	LOAD BALANCER IN ENVIRONMENT SERVICES PATTERNS
6,557,065 B1	December 20, 1999	CPU EXPANDABILITY BUS
6,529,198	March 15, 2000	PARALLEL RENDERING DEVICE
6,496,187	December 27, 1999	GRAPHICS SYSTEM CONFIGURED TO PERFORM PARALLEL SAMPLE TO PIXEL

		CALCULATION
6,292,200	October 23, 1998	APPARATUS AND METHOD FOR UTILIZING MULTIPLE RENDERING PIPES FOR A SINGLE 3-D DISPLAY
6,259,460 B1	March 26, 1998	METHOD FOR EFFICIENT HANDLING OF TEXTURE CACHE MISSES BY RECIRCULATION
6,212,617	June 30, 1998	PARALLEL PROCESSING METHOD AND SYSTEM USING A LAZY PARALLEL DATA TYPE TO REDUCE INTER-PROCESSOR COMMUNICATION
6,212,261 B1	August 19, 1999	INTERNET-BASED TELEPHONE CALL MANAGER
6,191,800	August 11, 1998	DYNAMIC BALANCING OF GRAPHICS WORKLOADS USING A TILING STRATEGY
6,118,462	September 9, 1997	COMPUTER SYSTEM CONTROLLER HAVING INTERNAL MEMORY AND EXTERNAL MEMORY CONTROL
5,841,444	March 21, 1997	MULTIPROCESSOR GRAPHICS SYSTEM
5,794,016	December 11, 1995	PARALLEL-PROCESSOR GRAPHICS ARCHITECTURE
5,757,385	January 30, 1996	METHOD AND APPARATUS FOR MANAGING MULTIPROCESSOR GRAPHICAL WORKLOAD DISTRIBUTION
5,745,762	December 15, 1994	ADVANCED GRAPHICS DRIVER ARCHITECTURE SUPPORTING MULTIPLE SYSTEM EMULATIONS
5,535,410	November 8, 1994	PARALLEL PROCESSOR HAVING DECODER FOR SELECTING SWITCH FROM THE GROUP OF SWITCHES AND CONCURRENTLY INPUTTING MIMD INSTRUCTIONS WHILE PERFORMING SIMD OPERATION
5,475,856	October 17, 1994	DYNAMIC MULTI-MODE PARALLEL PROCESSING ARRAY
2008/0143731	November 30, 2007	VIDEO RENDERING ACROSS A HIGH SPEED PERIPHERAL INTERCONNECT BUS

2008/0007559	June 30, 2006	APPARATUS, METHOD AND A COMPUTER PROGRAM PRODUCT FOR PROVIDING A UNIFIED GRAPHICS PIPELINE FOR STEREOSCOPIC RENDERING
2006/0290700 A1	September 1, 2006	MULTIPLE PARALLEL PROCESSOR COMPUTER GRAPHICS SYSTEM
2006/0274073 A1	November 17, 2004	MULTIPLE GRAPHICS ADAPTER CONNECTION SYSTEMS
2006/0156399	December 30, 2004	SYSTEM AND METHOD FOR IMPLEMENTING NETWORK SECURITY USING A SEQUESTERED PARTITION
2005/0190189	February 22, 2005	SYSTEM AND METHOD FOR GPU-BASED 3D NONRIDGID REGISTERATION
2005/0166207	December 27, 2004	SELF-OPTIMIZING COMPUTER SYSTEM
2005/0122330	November 12, 2004	SYSTEMS AND METHODS FOR DOWNLOADING ALGORITHIMIC ELEMENTS TO A COPROCESSOR AND CORRESPONDING TECHNIQUES
2004/0223003 A1	June 9, 2004	PARALLEL PIPELINED MERGE ENGINES
2004/0196289 A1	April 23, 2004	UPGRADING AN INTEGRATED GRAPHICS SUBSYSTEM
2003/0188075 A1	March 31, 2003	CPU EXPANDABILITY BUS
2003/0171907 A1	March 4, 2003	METHODS AND APPARATUS FOR OPTIMIZING APPLICATIONS ON CONFIGURABLE PROCESSORS
2003/0164834 A1	March 1, 2002	SYSTEM AND METHOD UTILIZING MULTIPLE PIPELINES TO RENDER GRAPHICAL DATA
2003/0164832 A1	March 4, 2002	GRAPHICAL DISPLAY SYSTEM AND METHOD
2003/0117971 A1	December 21, 2001	SYSTEM, METHOD, AND ARTICLE OF MANUFACTURE FOR PROFILING AN EXECUTABLE HARDWARE MODEL USING CALLS TO PROFILING FUNCTIONS
2002/0145612	January 29, 2002	METHOD AND SYSTEM FOR MINIMIZING

		AN AMOUNT OF DATA NEEDED TO TEST DATA AGAINST SUBAREA BOUNDARIES IN SPATIALLY COMPOSITED DIGITAL VIDEO
2002/0059302 A1	October 9, 2001	DATA COMMUNICATION SYSTEM AND METHOD, COMPUTER PROGRAM, AND RECORDING MEDIUM

FOREIGN PUBLICATIONS

NUMBER	PUBLICATION DATE	TITLE
WO 2004/070652 A2	August 19, 2004	METHOD AND SYSTEM FOR COMPOSITING THREE-DIMENSIONAL GRAPHICS IMAGES USING ASSOCIATED DECISION MECHANISM

INTERNATIONAL SEARCH REPORTS

App. No.	Filing Date
EP 04 79 9376	October 14, 2008
PCT/IB07/03464	September 22, 2008
PCT/US07/26466	July 16, 2008
PCT/IB06/01529	December 31, 2007
PCT/IL2004001069	June 30, 2005

TECHNICAL PUBLICATIONS

Publication by TW Crockett entitled, "An Introduction to Parallel Rendering", in Parallel Computing, 1997, Elsevier Science, 29 Pages.

Silicon Graphics, Inc. pdf. document entitled "OpenGL Multipipe™ SDK White Paper", 2002,2003,pages 1-32.

Silicon Graphics, Inc. online document entitled "Additional Information for: OpenGL Multipipe™ SDK White Paper (IRIX 6.5)", published February 1, 2003, 2 pages.

Technical publication by Li et al entiteled "ParVox—A Parallel Splatting Volume Rendering System for Distributed Visualization," October 1997, 7 Pages.

Department of Computer Science, University of North Carolina publication by Molnar et al. entitled, "PixelFlow: High-Speed Rendering Using Image Composition," 1992, 10 Pages.

ABSTRACTS OF DISCLOSURE

U.S. Patent No. 7,477,256 B1 to Johnson discloses a system and method for providing a dedicated digital interface between multiple graphics devices. The dedicated interface provides a point-to-point connection between each of the multiple graphics devices for the transfer of digital pixel data and synchronization signals. Graphics processing, including combining of portions of a displayable image, is distributed between the multiple graphics devices. One of the multiple graphics devices, a master graphics device converts the combined portions of the displayable image as needed for a specific display device.

U.S. Patent No. 7,372,465 to Tamasi et al. discloses a system and method that processes graphics data for remote display. A graphics processing system including a plurality of graphics processing devices is coupled to a host system that includes a host graphics processor and a display device that is remote relative to the graphics processing system. Graphics processing performance may be scaled by distributing processing between the plurality of graphics processing devices and the host graphics processor such that each of the plurality of graphics processing devices and the host graphics processor produces a portion of an image. The portions are combined to produce the image, which is output by the host graphics processor to the display device.

U.S. Patent No. 7,325,086 B2 to Kong et al. discloses supporting multiple graphics processing units (GPUs) comprising a first path coupled to a north bridge device (or a root complex device) and a first GPU, which may include a portion of the first GPU's total communication lanes. A second communication path may be coupled to the north bridge device and a second GPU and may include a portion of the second GPU's total communication lanes. A third communication path may be coupled between the first and second GPUs directly or through one or more switches that can be configured for single or multiple GPU operations. The third communication path may include some or all of the remaining communication lanes for the first and second GPUs may each utilize an 8-lane PCI express communication path with the north bridge device and an 8-lane PCI express communication path with each other.

U.S. Patent No. 7,324,547 B1 to Alfieri et al. discloses a novel network architecture that integrates the functions of an internet protocol (IP) router into a network processing unit (NPU) that resides in a host computer's chipset such that the host computer's resources are perceived as separate network appliances. The NPU appears logically separate from the host computer even though, in one embodiment, it is sharing the same chip.

U.S. Patent No. 7,324,111 B2 to Diamond discloses one embodiment of a connector for a stand-alone graphics module, adapted for coupling a computing device to the stand-alone graphics module, which is external to the computing device. The connector is adapted for receiving a PCI express signal from the computing device and for delivering the PCI express signal to the stand-alone graphics module.

The connector is further adapted for receiving display output signals from the stand-alone graphics module and delivering the display output signals to the computing system, e.g., for use in accordance with one or more output display panels coupled to said computing device.

U.S. Patent No. 7,119,808 to Gonzalezz et al. discloses an accelerated graphics processing subsystem that significantly increases the processing speed of computer graphics commands. The preferred embodiment of this invention presents a first-of-its-kind graphics processing subsystem that combines the processing power of multiple, off-the-shelf, video cards, each one having one or more graphic processor units. The video cards can be used without substantial modification. Under the preferred embodiment, each video card processes instructions for drawing a predetermined portion of the screen which is displayed to the used through a monitor or other visual output device. The invention harnesses the power of multiple video cards without suffering from the high bandwidth constraints affecting prior attempts at parallel graphics processing subsystems.

U.S. Patent No. 7,051,139 to Peleg et al. discloses an invention's embodiments provide a computer system with a high speed, high bandwidth expandability bus for integrated and non-integrated CPU products. The computer system includes a processor, a chipset coupled to the processor, a graphics processor coupled to the chipset for controlling a video display and a main memory coupled to the chipset. The computer system further includes an expandability bus, which is coupled at one end to the chipset and at the other end to a replaceable electronic component. The expandability bus can be changeably configured to enable or disable bus mastering at both ends, as required, to operate with whichever replaceable electronic component is installed.

U.S. Patent No. 6,789,154 to Lee et al. discloses a system for providing video, the system having a system bus, which in one embodiment is and Advanced Graphics Port (AGP) bus. The system bus is connected to a data bridge, which is connected to a second and third AGP bus. Each of the AGP busses are connected to graphics processors. The bridge routes data requests from one graphics processor to the second graphics processor without accessing the system AGP bus based upon memory mapping information stored in a routing table or a register set. In another aspect of the present invention, the bridge responds to initialization requests using attributes that may vary depending on the specific mode of operation. Another aspect of the present invention allows for conversion between various AGP protocol portions.

U.S. Patent No. 6,885,376 to Tang-Petersen et al. discloses a system, method, and computer program for creating a sequence of computer graphics frames, using a plurality of rendering pipelines. For each frame, each rendering pipeline receives a subset of the total amount of graphics data for the particular frame. At the completion of a frame, each rendering pipeline sends a performance report to a performance monitor. The performance monitor determines whether or not there was a significant disparity in the time required by the respective rendering pipelines to render their tiles. If a disparity is detected, and if the disparity is determined to be greater than some threshold, an allocation module resizes the tiles for the next frame. This serves to balance the load across rendering pipelines for each frame.

U.S. Patent No. 6,864,894 to Naya et al. discloses a plurality of measuring units each comprising a dielectric block, a metal film layer which is formed on a surface of the dielectric block and a sample holder are supported on a support. The support is moved by a support drive means to bring in sequence the measuring units to a measuring portion comprising an optical system which projects a light beam emitted from a light source, and a photodetector which detects attenuation in total internal reflection by detecting the intensity of the light beam which is reflected in total internal reflection at the interface between the dielectric block and the metal film layer. In this measuring apparatus, lots of samples can be measured in a short time.

U.S. Patent No. 6,844,879 to Miyauchi discloses a drawing apparatus which includes a display element on which images are displayed and a plurality of loosely coupled drawing elements each of which executes drawing processing in parallel. At an update time predetermined for a display screen of the display element, each of the drawing elements updates the display screen only if none of the drawing elements is executing drawing processing. A parallel drawing method of the present invention executed by a plurality of drawing elements, each of which executes drawing processing, includes; receiving a drawing instruction; and updating the display screen of the display element by each of the drawing elements only if none of the drawing elements is executing drawing processing at an update time predetermined for the display screen.

U.S. Patent No. 6,842,180 B1 to Maiyuran et al. discloses an electronic device that has an integrated central processing unit (CPU) including a pre-fetch stride analyzer and an out-of-order engine. The electronic device also has a graphics engine, having graphics memory, that is coupled to the integrated CPU. A main memory that is coupled to a memory controller is provided. The memory controller is also coupled to the CPU and the graphics engine. The device has a host address decoder coupled to the integrated CPU. A front side bus (FSB) is provided that is coupled to the integrated CPU and the host address decoder. Also provided is a plurality of memory components. Accordingly, either the plurality of memory components or the graphics memory can be shared to perform alternate memory functions. Additionally, a method is provided that determines allocation availability between memory components in an integrated computer processing unit. The method also shares an available memory component as a pre-fetch buffer and another available memory component as a victim cache.

U.S. Patent No. 6,801,202 B2 to Nelson et al. discloses a method and computer graphics system capable of implementing multiple pipelines for the parallel processing of graphics data. For certain data, a requirement may exist that the data be processed in order. The graphics system may use a set of tokens to reliably switch between ordered and unordered data modes. Furthermore, the graphics system may be capable of super-sampling and performing real-time convolution. In one embodiment, the computer graphics system may comprise a graphics processor, a sample buffer, and a sample-to-pixel calculation unit. The graphics processor may be configured to receive graphics data and to generate a plurality of samples for each of a plurality of frames. The sample buffer, which is coupled to the graphics processor, may be configured to store the samples. The sample-to-pixel calculation unit is programmable to generate a plurality of output pixels by filtering the rendered samples using a filter.

U.S. Patent No. 6,778,177 to Furtner discloses a method for rasterizing a graphic primitive in a graphics system generates, starting from graphic primitive description data, pixel data for the graphic primitive, the graphics system comprising a memory which is divided up into a plurality of blocks (a, a+1, b, b+1) which are each associated with a predetermined one of a plurality of areas of mapping screen. Each block of the plurality of blocks (a, a+1, b, b+1) is associated with a memory page in the memory. The method includes scanning the pixels associated with the graphic primitive in one of the plurality of blocks into which the graphic primitive extends, repeating the preceding steps until all the pixels associated with the graphic primitive have been scanned in each of the plurality of blocks into which the graphic primitive extends, and outputting the pixel data.

U.S. Patent No. 6,753,878 to Heirich et al. discloses an image generator organized into a plurality of rending engines, each of which renders an image of a part scene and provides the part image to a merge engine associated with that rendering engine. The image is a part image in that is usually contains less than all of the objects in the image to be rendered. The merge engine merges the part image from its associated rendering engine with the part image provided by a prior merge engine and provides the merged part image to the next merge engine. One ore more merge engines are designated to output merge engines and these output merge engines output a merged part image that is (a portion of) the ultimate output of the image generator, the full rendered image. Each merge engine performs its merge process on

the pixels it has from its rendering engine and from its prior neighbor merge engine, in a pipelined manner and without necessarily waiting for all of the pixels of the part image or the merged part image.

U.S. Patent No. 6,741,243 to Lewis et al. discloses a method and system for providing a graphical image on a display of a system. The graphical image is provided from data describing a plurality of primitives. The display includes a plurality of pixels. The method and system include providing a plurality of variable-sized bins containing the plurality of primitives and rendering the plurality of primitives by rendering each of the plurality of variable-sized bins variable-sized bin by variable-sized bin.

U.S. Patent No. 6,683,614 to Walls et al. discloses a system and method for automatically configuring graphics pipelines by tracking a region of interest in a computer graphical display system. The method comprises receiving updated definitional information on a selected region of interest of a display device of the computer graphical display system in response to a change in definition of the selected region of interest and automatically configuring the plurality of graphics pipelines relative to the selected region of interest based at least in part on the updated definitional information.

U.S. Patent No. 6,578,068 to Bowman-Amuah discloses a system and method for distributing incoming requests from a user interface amongst a client and server components for optimizing usage of resources. Incoming requests are first received and stored by an activity module. The activity module instructs a client to handle a first subset of the requests and passes a second subset of the request on to a utilization-based load balancer. The second subset of the requests are stored on the load balancer, and an availability of server components is determined and a listing of available server components is compiled. A determination is made as to which server component on the listing of available server components is most appropriate to receive a particular request. Each particular request of the second subset of requests is sent to the selected server component determined to be most appropriate to receive the particular request.

U.S. Patent No. 6,557,065 B1 to Peleg et al. discloses embodiments of the invention which provide a computer system with a high speed, high bandwidth expandability bus for integrated and non-integrated CPU products. The computer system includes a processor, a chipset coupled to the processor, a graphics processor coupled to the chipset for controlling a video display and a main memory coupled to the chipset. The computer system further includes an expandability bus, which is coupled at one end to the chipset and at the other end to a replaceable electronic component. The expandability bus can be changeably configured to enable or disable bus mastering at both ends, as required, to operate with whichever replaceable electronic component is installed.

U.S. Patent No. 6,529,198 to Miyauchi discloses a rendering command/data generator that distributes rendering commands and data to each of rendering devices 3-1-3-n with rendering commands and data for one screen as a unit. Each of rendering devices 3-1-3-n carries out the generating of display data and storing of the display data in a rendering memory incorporated in each rendering device in accordance with the rendering commands and data. The content of the rendering memories is read out by a read signal that: is supplied from display control unit 2 and synchronized with the scan of display 7. Window number buffer 4 issues the window number of the window in which a pixel currently to be displayed is included. Window number/rendering device management table 6 issues the device number of the rendering device as a selection signal. Display switch 5 selects the rendering device of the device number indicated by the selection signal to connect the rendering device to the display. In this way, the most recent display data of the window of the above-described window number is supplied to the display.

U.S. Patent No. 6,496,187 to Deering et al. discloses a graphics system that is configured to utilize a sample buffer and a plurality of parallel sample-to-pixel calculation units, wherein the sample-pixel calculation units are configured to access different portions of the sample buffer in parallel. The

graphics system may include a graphics processor, a sample buffer, and a plurality of sample-to-pixel calculation units. The graphics processor is configured to receive a set of three-dimensional graphics data and render a plurality of samples based on the graphics data. The sample buffer is configured to store the plurality of samples for the sample-to-pixel calculation units, which are configured to receive and filter samples from the sample buffer to create output pixels. Each of the sample-to-pixel calculation units are configured to generate pixels corresponding to a different region of the image. The region may be a vertical or horizontal stripe of the image, or a rectangular portion of the image. Each region may overlap the other regions of the image to prevent visual aberrations.

U.S. Patent No. 6,292,200 to Bowen et al. discloses a computer graphics system having a hyperpipe architecture. Multiple rendering pipes are coupled together through a hyperpipe network scheme. Each of the rendering pipes are capable of rendering primitives for an entire frame or portions thereof. This enables multiple rendering pipes to process graphics data at the same time. A controller coordinates the multiple rendering pipes by sending requests to the appropriate rendering pipes to retrieve the pixel data generated by that particular pipe. It then merges the pixel data received from the various rendering pipes. A single driver then draws the three-dimensional image out for display.

U.S. Patent No. 6,259,460 B1 to Gossett et al. discloses a method of a computer graphics system which recirculates texture cache misses into a graphics pipeline without stalling the graphics pipeline, increasing the processing speed of the computer graphics system. The method reads data from a texture cache memory by a read request placed in the graphics pipeline sequence, then reads the data from the texture cache memory and places the data in the pipeline sequence. If the data is not stored in the texture cache memory, the method recirculates the read request in the pipeline sequence by indicating in the pipeline sequence that the data is not stored in the texture cache memory, placing the read request at a subsequent, determined place in the pipeline sequence, reading the data into the texture cache memory from a main memory, and executing the read request from the subsequent, determined place and after the data has been read into the texture cache memory.

U.S. Patent No. 6,212,617 to Hardwick discloses a parallel programming system which provides a lazy collection oriented data type that reduces inter-processor communication in programs executed on parallel computers. The lazy collection oriented data type is provided as a data type in a parallel programming language. The parallel language supports both data-parallel and control-parallel operations. These operations take advantage of the lazy collection oriented data type to defer or reduce inter-processor communication until an operation on the data type requires that it be balanced across a set of processors.

U.S. Patent No. 6,212,261 B1 Meubus et al. a method that allows data access service provider subscribers to manage their telephone service through a data connection. The subscriber is enabled to obtain call data information and is provided real time control. During a data call, a visual incoming call indicator informs the subscriber, through a popup window, connected to the data access service provider that there is a call attempt. A visual message waiting indicator allows a subscriber, connected to the data access service provider to be notified of a pending message on the voice message system. A visual call disposition allows the subscriber, through the data connection, to dispose of calls. The call disposition options include forwarding a call to voice mail, playing an announcement to the calling party, forwarding the call to another line, sending a text message which could be converted to speech using text to speech technology, answering the call using voice over data call or terminating the data connection in order to accept the call.

U.S. Patent No. 6,191,800 to Arenburg et al. discloses a method of generating graphic images on a display device of a computer system, by dividing the viewable area of the display device into a plurality of tiles, assigning each of the tiles to respective rendering processes, rendering a frame on the display

device using the rendering processes, and adjusting the sizes of the tiles in response to the rendering step. The method then uses the adjusted tile sizes to render the next frame. The tile areas are preferably constrained from becoming too small. The adjusted tile sizes may optionally be smoothed, such as by averaging in weighted sizes from previous frames. The tile sizes are adjusted by measuring the rendering times required for each rendering process to complete a respective portion of the frame, and then resizing the tiles based on the measured rendering times.

U.S. Patent No. 6,118,462 to Margulis discloses an invention relating generally to an optimized memory architecture for computer systems and, more particularly, to integrated circuits that implement a memory subsystem that is comprised of internal memory and control for external memory. The invention includes one or more shared high-bandwidth memory subsystems, each coupled over a plurality of buses to a display subsystem, a central processing unit (CPU) subsystem, input/output (I/O) buses and other controllers. Additional buffers and multiplexers are used for the subsystems to further optimize system performance.

U.S. Patent No. 5,841,444 to Mun et al. discloses a multiprocessor graphics system having a pixel link architecture, which includes: 1) a plurality of sub-graphics systems each of which assigned to each of a plurality of sub-screens provided by sectioning a display screen; and 2) a ring network for connecting the plurality of sub-graphics systems. Each of the sub-graphics systems includes a geometry engine, a raster engine, a local frame buffer and a pixel distributor. An interconnection network bottleneck between the raster engine and frame buffer is removed and a conventional memory system can be used by reducing the number of data transmissions between the raster engine and frame buffer while maintaining image parallelism and object parallelism.

U.S. Patent No. 5,794,016 to Kelleher discloses a parallel-processor graphics architecture appropriate for multimedia graphics workstations that are scalable to the needs of a user. The graphics architecture includes one or more rendering processors and a graphics memory that is partitioned into blocks. Noncontiguous groups of the blocks are then assigned to different processors. The parallel-processors graphics architecture is scalable by the number of rendering processors utilized, and is configurable with respect to the allocation of the groups of the blocks to specific rendering processors.

U.S. Patent No. 5,757,385 to Narayanaswami et al. discloses an apparatus for utilizing multiple processors to render graphical objects for display including an apparatus for storing in memory a list of pixel locations assigned to each of the processors, apparatus for scan converts each received graphical object into pixels, and each processor including apparatus for rendering graphical object pixels at pixel locations assigned to the processor. In addition, a method of utilizing multiple processors to render graphical objects for display including the steps of storing in memory a list of pixel locations assigned to each of the processors, scan converting each received graphical object into pixels, and each processor rendering graphical object pixels aat pixel locations assigned to the processor.

U.S. Patent No. 5,745,762 to Celi, Jr. et al. discloses a support architecture that facilitates use of display device drivers containing a minimum of hardware-specific software code. A driver need support only a relatively few common functions, which act as building blocks for the larger, more complex operations typically requested by graphics engines. In order to mediate between the limited-instruction-set device driver and the various higher-level graphics engines, the invention includes a series of translation modules that simplify engine-originated instructions into simpler graphic components. A video manager supervises routing of instructions to the specific drivers they designate, and serializes access to hardware components so that graphic commands execute atomically (i.e., without interruption). The invention can accommodate multiple device drivers in parallel or serial configurations. Device drivers may be paired with specific graphics adapters, or multiple drivers, each responsive to a different graphics engine, can share a single graphics adapter. A particular driver may be selected by an application program, by a

graphics engine, by a translation module, or by the video manager based on considerations of compatibility and efficiency. Ordinarily, a device driver controls the entire video display or a discrete region thereof during the time it processes a particular graphic function; however, the invention can also accommodate multiple video displays, each controlled by a separate device driver.

U.S. Patent No. 5,535,410 to Watanabe et al. discloses a parallel processor which utilizes a memory cell array for rapidly performing parallel processing by switching between SIMD and MIMD operations depending on the type of problems to be solved. Where SIMD and MIMD operations are mixed in an application, the time loss in the switching therebetween is eliminated so as to enhance the speed of the processing. The parallel processor comprises a two-dimensional memory array for storing data to be operated on; a transfer network for transferring to a group of processing elements the data read in parallel from word lines connected to memory cells in the two-dimensional memory array, the group of processing elements performing parallel processing on the data transferred thereto; signal lines for transmitting an instruction in a SIMD operation mode; an instruction buffer for storing and forwarding parallelly instructions in a MIMD operation mode; and a group of switches for switching between the SIMD and the MIMD operation mode.

U.S. Patent No. 5,475,856 to Kogge discloses a Parallel RISC computer system provided by a multi-mode dynamic multi-mode parallel processor array with one embodiment illustrating a tightly coupled VLSI embodiment with an architecture which can be extended to more widely placed processing elements through the interconnection network which couples multiple processors capable of MIMD mode processing to one another with broadcast of instructions to selected groups of units controlled by a controlling processor. The coupling of the processing elements logic enables dynamic mode assignment and dynamic mode switching, allowing processors operating in a SIMD mode to make maximum memory and cycle time usage. On and instruction by instruction level basis, modes can be switched from SIMD to MIMD, and even into SISD mode on the controlling processor for inherently sequential computation allowing a programmer or complier to build a program for the computer system which uses the optimal kind of parallelism (SISD, SIMD, MIMD). Furthermore, this execution, particularly in the SIMD mode, can be set up for running applications at the limit of memory cycle time. With the ALLNODE switch and alternatives paths a system can be dynamically achieved in a few cycles for many many processors. Each processing element and memory and has MIMD capability the processor's an instruction register, condition register and program counter provide common resources which are used in MIMD and SIMD. The program counter become a base register in SIMD mode.

U.S. Publication No. 2008/0143731 to Cheng et al. discloses graphics generated by one graphics processor which are transferred across a high speed interconnect bus to a frame buffer. The rendered frames from the frame buffer are presented on a display by way of a display interface in communication with the frame buffer. The display interface of another existing (e.g. integrated) graphics adapter/subsystem may be used to present the rendered frames on an interconnected display.

U.S. Publication No. 2008/0007559 to Kalaiah et al. discloses a device for rendering to multiple viewpoints of a stereoscopic display. The device includes vertex shaders which receive vertices corresponding to primitives and process viewpoint dependent information. The device also includes a primitive replication unit which replicates primitives according to a number of viewpoints supported by the stereoscopic display. The primitive replication unit adds unique view tags to each of the primitives which identify the viewpoint that the respective primitive is destined for. Each replicated primitive is processed by a rasterizer and converted into pixels. The rasterizer adds a view tag to the rasterized pixels so that the pixels identify a respective primitive and identify a respective pixel buffer that the pixel is destined for. The pixels can then be processed by a pixel processing unit and written to a pixel buffer corresponding to a respective viewpoint. The pixels are subsequently output to the stereoscopic display.

U.S. Publication No. 2006/0290700 A1 to Gonzalez et al. discloses an accelerated graphics processing subsystem that significantly increases the processing speed of computer graphics commands. The preferred embodiment of this invention presents a first-of-its-kind graphics processing subsystem that combines the processing power of multiple, off-the-shelf, video cards, each one having one or more graphic processor units. The video cards can be used without substantial modification. Under the preferred embodiment, each video card processes instructions for drawing a predetermined portion of the screen which is displayed to the user through a monitor or other visual output device. The invention harnesses the power of multiple video cards without suffering from the high bandwidth constraints affecting prior attempts at parallel graphics processing subsystems.

U.S. Publication No. 2006/0274073 A1 to Johnson et al. discloses a system providing a digital multi-bit connection between two or more graphics adapters. Each graphics adapter is manufactured as a printed circuit board including a finger-type edge connector. When two or more graphics adapters are installed in a system the edge connectors of each graphics adapter may be coupled to each other via a connection device that provides a portion of the digital multi-bit connection. The remainder of the digital multi-bit connection is provided by conductive traces coupling each finger of the edge connector to a graphics processing unit that is affixed to the graphics adapter. The connection device may be installed by an end-user as each additional graphics adapter is installed in the system.

U.S. Publication No. 2006/0156399 to Parmer et al. discloses a system and method which are implemented within a computing system to perform tamper-resistant network security operations. For example, a method of one embodiment comprises: sequestering a partition on the computing system, the partition including a region of memory and a logical or physical processing element; forwarding incoming and/or outgoing data traffic through the sequestered portion, the incoming data traffic being received by the computing system from a network and the outgoing data traffic being transmitted from the computing system over the network; performing on ore more security operations on the data traffic within the sequestered partition.

U.S. Publication No. 2005/0190189 A1 to Chefd'hotel et al. discloses a method of registering two images using a graphics processing unit which includes providing a pair of images with a first and second image, calculating a gradient of the second image, initializing a displacement field on the grid point domain of the pair images, generating textures for the first image, the second image, the gradient, and the displacement field, and loading said textures into the graphics processing unit. A pixel buffer is created and initialized with the texture containing the displacement field. The displacement field is updated from the first image, the second image, and the gradient for one or more iterations in one or more rendering passes performed by the graphics processing unit.

U.S. Publication No. 2005/0166207 to Baba et al. discloses a self-optimizing computer system that can achieve ultimate optimization (improvement in the speed) by preparing a mechanism that can observe the behavior of the program execution in the self-optimizing computer system and optimize dynamically depending on the execution behavior of program. The self-optimizing computer system comprising multiple processing units, characterized in that each of the processing units operates as at least one of an operation processing unit for executing a program, an observation processing unit for observing the behavior of the program under execution, an optimization processing unit for performing an optimization process according to the observation result of the observation processing unit, and a resource management processing unit for performing a resource management process of whole of the system such as a change of the contents of execution.

U.S. Publication No. 2005/0122330 to Boyd et al. discloses systems and methods for downloading algorithmic elements to a coprocessor and corresponding processing and communication techniques are provided. For an improved graphics pipeline, the invention provides a class of co-

processing device, such as a graphics processor unit (GPU), providing improved capabilities for an abstract or virtual machine for performing graphics calculations and rendering. The invention allows for runtime-predicated flow control of programs downloaded to coprocessors, enables coprocessors to include indexable arrays of on-chip storage elements that are readable and writable during execution of programs, provides native support for textures and texture maps and corresponding operations in a vertex shader, provides frequency division of vertex streams input to a vertex shader with optional support for a stream modulo value, provides a register storage element on a pixel shader and associated interfaces for storage associated with representing the "face" of a pixel, provides vertex shaders and pixel shaders with more on-chip register storage and the ability to receive larger programs than any existing vertex or pixel shaders and provides 32 bit float number support in both vertex and pixel shaders.

U.S. Publication No. 2004/0223003 A1 to Heirich et al. discloses an image generator organized into a plurality of rendering engines, each of which renders an image of a part scene and provides the part image to a merge engine associated with that rendering engine. The image is a part image in that it usually contains less than all of the objects in the image to be rendered. The merge engine merges the part image from its associated rendering engine with the part image provided by a prior merge engine and provides the merged part image to a next merge engine. One or more merge engines are designated the output merge engines and these output merge engines output a merged part image that is (a portion of) the ultimate output of the image generator, the full rendered image. Each merge engine performs its merge process on the pixels it has from its rendering engine and from its prior neighbor merge engine, in a pipelined manner and without necessarily waiting for all of the pixels of the part image or the merged part image.

U.S. Publication No. 2004/0196289 A1 to Langendorf et al. discloses an apparatus and methods for allowing two graphics controllers to cooperate on a single screen and for modifying the AGP protocol to provide symmetric capabilities for both AGP targets and AGP masters. According to one embodiment of the present invention two graphics controllers may cooperate as one virtual graphics controller. A first graphics controller renders a first subset of pixels of a display to a local memory of the first graphics controller. A second graphics controller renders a second subset of pixels of the display to a local memory of the second graphics controller. Then, after both the first graphics controller and the second graphics controller have completed their respective rendering, merging the content of the local memory of the first graphics controller and the content of the local memory of the second graphics controller.

U.S. Publication No. 2003/0188075 A1 to Peleg et al. discloses a computer system with a high speed, high bandwidth expandability bus for integrated and non-integrated CPU products. The computer system includes a processor, a chipset coupled to the processor, a graphics processor coupled to the chipset for controlling a video display and a main memory coupled to the chipset. The computer system further includes an expandability bus, which is coupled at one end to the chipset and at the other end to a replaceable electronic component. The expandability bus can be changeably configured to enable or disable bus mastering at both ends, as required, to operate with whichever replaceable electronic component is installed.

U.S. Publication No. 2003/0171907 A1 to Gal-On et al. discloses methods and apparatus directed to optimizing configurable processors to assist a designer in efficiently matching a design of an application and a design of a processor. In one aspect, methods and apparatus according to the present invention optimize a hardware architecture having one or more application specific processors. The methods and apparatus include modeling one or more of the application specific processors to generate a simulated hardware architecture and analyzing a compiled program for the simulated hardware architecture to determine one or more resource parameters for one or more program sections of the compiled program. The methods and apparatus provide one or more suggestions for modifying one or more of the application specific processors and the program sections in response to the resource

parameter to optimize one or both of the compiled program and the hardware architecture.

U.S. Publication No. 2003/0164834 A1 to Lefebvre et al. discloses a system for rendering graphical data utilizing a plurality of frame buffers, a plurality of graphics pipelines, a compositor, and logic. The plurality of graphics pipelines are configured to receive graphics commands and to render graphical data to each of the plurality of frame buffers based on the received graphics commands. The compositor is configured to receive a control signal and to interface the graphical data with a display device based on the control signal. The logic is configured to analyze the graphics commands and to make a determination, based on the graphics commands, as to which pixels defined by the graphical data are associated with three-dimensional (3D) regions. The logic is further configured to transmit the control signal to the compositor, wherein the control signal is based on the determination.

U.S. Publication No. 2003/0164832 A1 to Alcorn discloses a graphical display system utilizing a plurality of graphics pipelines, a compositor, and application interface logic. The plurality of graphics pipelines are configured to render graphical data in parallel. The compositor is configured to define a composite data signal that is based on the graphical data rendered by each of the pipelines. The application interface logic is configured to retrieve configuration data indicative of a configuration of the compositor. The application interface logic is further configured to provide the configuration data to a graphics application, wherein the graphics application is configured to provide graphical data to the plurality of pipelines based on the configuration data.

U.S. Publication No. 2003/0117971 A1 to Aubury discloses a system, method and article of manufacture provided for profiling an executable hardware model selecting a plurality of profiling functions of a profiling process. An application having application functions targeted for implementation in reconfigurable logic is preprocessed for inserting calls to the profiling functions. The application is executed. A profile is generated based on the profiling functions called during execution of the application.

U.S. Publication No. 2002/0145612 to Blythe et al. discloses a method and system for minimizing an amount of data needed to test data against subarea boundaries in spatially composited digital video. Spatial compositing uses a graphics unit or pipeline to render a portion (subarea) of each overall frame of digital video images. This reduces the amount of data that each processor must act on and increases the rate at which an overall frame is rendered. Optimization of spatial compositing depends on balancing the processing load among the different pipelines. The processing load typically is direct function of the size of a given subarea and a function of the rendering complexity for objects within this subarea. Load balancing strives to measure these variables and adjust, from frame to frame, the number, sizes, and positions of the subareas. The cost of this approach is the necessity to communicate, in conjunction with each frame, the graphics data that will be rendered. Graphics data for a frame is composed of geometry chunks. Each geometry chunk is defined by its own bounding region, where the bounding region defines the space the geometry chunk occupies on the compositing window. Only the parameters that define the bounding region are communicated to each graphics unit in conjunction with the determination of which graphics unit will render the geometry chunk defined by the bounding region. The actual graphics data that comprises the geometry chunk is communicated only to those geometry units that will actually render the geometry chunk. This reduces the amount of data needed to communicate graphics data information in spatially composited digital video.

U.S. Publication No. 2002/0059302 A1 to Ebihara discloses an invention providing data communication technology for improving the efficiency of cooperation of two or more information processing units (GSM) for more sophisticated processing. According to the present invention, there are provided four GSMs, a sub-MG (merger) for merging data output from the GSMs, and a main MG for merging data output from the four sub-MGs. Data output from the GSMs are stored in parallel in a

register on a unit length basis. Then the data stored in the register are serially read on the unit length basis to form serial data. When the serial data contain altered data, auxiliary data for identifying which data have been altered or modified are added to a predetermined portion of the serial data. Then the serial data with the auxiliary data added thereto are output to the main MG. On the other hand, parallel data to be output from a main SYNC to each GSM are copied, and the copies of the same parallel data are propagated over all the GSMs at the same time

WIPO Publication No. 2004/070652 A2 describes a method and system for compositing a plurality of three-dimensional Sub-images by examining the Depth values of the Pixels corresponding to same spatial location in each Sub-Image and compositing the content of the Pixel having the greatest Depth value. The Depth values are divided into two or more binary Segments, where the bit length of the Segments is determined according to their level of significance. In a first step, the numerical values of the Segments having the same level of Significance are simultaneously compared, and accordingly a group designating the Depth values which the numerical value of their Most Significant Segment is the greatest is determined, and a Grade is evaluated for the Least Significant Segments indicating their numerical size in comparison with the numerical value of the other Segments of the same level of significance. In a second step, the Grades of the Segments of the Depth values which corresponds to the group are compared, and Depth value indications are removed from the group if their Grade is less than the highest Grade which corresponds to another Depth values group. The second step is repeated until the last set of Segments is reached or until a single Depth value is designated by the group.

Scientific publication entitled "An Introduction to Parallel Rendering," by TW Crockett describes how in computer graphics, rendering is the process by which an abstract description of a scene is converted to an image. When the scene is complex, or when high quality images or high frame rates are required, the rendering process becomes computationally demanding. To provide the necessary levels of performance, parallel computing techniques must be brought to bear. Today, parallel hardware is routinely used in graphics workstations, and numerous software-based rendering systems have been developed for general-purpose parallel architectures. This article provides an overview of the parallel rendering field, encompassing both hardware and software systems. The focus is on the underlying concepts and the issues which arise in the design of parallel renderers. We examine the different types of parallelism and how they can be applied in rending applications. Concepts from parallel computing, such as data decomposition and load balancing, are considered in relation to the rendering problem. Our survey explores a number of practical considerations as well, including the choice of architectural platform, communication and memory requirements, and the problem of image assembly and display. We illustrate the discussion with numerous examples from the parallel rendering literature, representing most of the principal rendering methods currently used in computer graphics.

Publication entitled "OpenGL Multipipe™ SDK White Paper" describes an application programming interface (API) designed to help software developers meet the demands of these new immersive environments. This product enables the application to take advantage of the scalability provided by additional pipes and other scalable graphics hardware, as well as to support immersive environments.

Online web-page entitled "Additional Information for: OpenGL Multipipe SDK White Paper (IRIX 6.5)" describes an application programming interface (API) designed to help software developers meet the demands of these new immersive environments. This product enables the application to take advantage of the scalability provided by additional pipes and other scalable graphics hardware, as well as to support immersive environments. OpenGL Multipipe SDK provides the following specific features: run-time configurability, run-time scalability, integrated support for scalable graphics hardware, and integrated support for stereo and immersive environments.

Publication entitled "ParVox- A Parallel Splatting Volume Rendering System for Distributed Visualization," describes a parallel volume rendering system, ParVox, for large volumes of 4-D data sets in regular structured grids. A parallel volume rendering API based on the splatting algorithm constitutes the core of the ParVox system. A network interface program takes commands from an X Window based GUI, calls the API to perform the rendering functions, compresses the rendered images and sends them back to the GUI window. The ParVox system is designed for interactive, distributed visualization of large multiple time steps, multiple parameters volume datasets. The parallel splatting algorithm employs no other object space decomposition and image space decomposition; an asynchronous image compositing scheme based on the direct send model reduces both the communication overhead and the synchronization overhead. The ParVox system architecture, the parallel algorithm and its implementation on the Cray T3D and the parallel wavelet compression algorithm are discussed extensively in this paper. The performance results and some optimization techniques are also presented.

Publication entitled "PixelFlow: High-Speed Rendering Using Image Composition," describes PixelFlow, an architecture for high-speed image generation that overcomes the transformation and frame buffer access bottlenecks of conventional hardware rendering architectures. PixelFlow uses the technique of image composition: it distributes the rendering task over an array of identical renderers, each of which computes a full-screen image of a fraction of the primitives. A high-performance image-composition network composites these images in real time to produce an image of the entire scene. Image-composition architectures offer performance that scales linearly with the number of renderers; there is no fundamental limit to the maximum performance achievable using this approach. A single PixelFlow renderer rasterizes up to 1.4 million triangles per second, and an *n* times this basic rate. PixelFlow performs antialiasing by supersampling. It supports deferred shading with separate hardware shaders that operate on composited images containing intermediate pixel data. PixelFlow shaders compute complex shading algorithms and procedural and image-based textures in real-time. The shading rate is independent of scene complexity. A PixelFlow system can be coupled to a parallel supercomputer to serve as an immediate mode graphics server, or it can maintain a display list for retained mode rendering.

A separate listing of the above references on PTO Form 1449 as well as hard copies of all non-U.S. references have been enclosed herewith for the convenience of the Examiner.

The Commissioner is hereby authorized to charge the requisite fee of $180.00, as well as any fee deficiencies or overpayments to Deposit Account No. 16-1340. A duplicate copy of this page is enclosed herewith.

Respectfully submitted,

Dated: December 8, 2009

Thomas J. Perkowski
Reg. No. 33,134
Attorney for Applicants
Thomas J. Perkowski, Esq., P.C.
22 Thornal Circle
Darien, Connecticut 06820
203-357-1950
http://www.tjpatlaw.com

CERTIFICATE OF MAILING UNDER
37 C.F.R. 1.08

I hereby certify that this correspondence
is being deposited with
the United States Postal Service
on December 8, 2009
in a Postage Prepaid envelope as
First Class Mail, addressed to:

Commissioner of Patents
P.O. Box 1450
Alexandria VA 22313-1450

Thomas J. Perkowski, Esq.
Reg. 33,134
Date: December 8, 2009

Publication entitled "ParVox- A Parallel Splatting Volume Rendering System for Distributed Visualization," describes a parallel volume rendering system, ParVox, for large volumes of 4-D data sets in regular structured grids. A parallel volume rendering API based on the splatting algorithm constitutes the core of the ParVox system. A network interface program takes commands from an X Window based GUI, calls the API to perform the rendering functions, compresses the rendered images and sends them back to the GUI window. The ParVox system is designed for interactive, distributed visualization of large multiple time steps, multiple parameters volume datasets. The parallel splatting algorithm employs no other object space decomposition and image space decomposition; an asynchronous image compositing scheme based on the direct send model reduces both the communication overhead and the synchronization overhead. The ParVox system architecture, the parallel algorithm and its implementation on the Cray T3D and the parallel wavelet compression algorithm are discussed extensively in this paper. The performance results and some optimization techniques are also presented.

Publication entitled "PixelFlow: High-Speed Rendering Using Image Composition," describes PixelFlow, an architecture for high-speed image generation that overcomes the transformation and frame buffer access bottlenecks of conventional hardware rendering architectures. PixelFlow uses the technique of image composition: it distributes the rendering task over an array of identical renderers, each of which computes a full-screen image of a fraction of the primitives. A high-performance image-composition network composites these images in real time to produce an image of the entire scene. Image-composition architectures offer performance that scales linearly with the number of renderers; there is no fundamental limit to the maximum performance achievable using this approach. A single PixelFlow renderer rasterizes up to 1.4 million triangles per second, and an *n* times this basic rate. PixelFlow performs antialiasing by supersampling. It supports deferred shading with separate hardware shaders that operate on composited images containing intermediate pixel data. PixelFlow shaders compute complex shading algorithms and procedural and image-based textures in real-time. The shading rate is independent of scene complexity. A PixelFlow system can be coupled to a parallel supercomputer to serve as an immediate mode graphics server, or it can maintain a display list for retained mode rendering.

A separate listing of the above references on PTO Form 1449 as well as hard copies of all non-U.S. references have been enclosed herewith for the convenience of the Examiner.

The Commissioner is hereby authorized to charge the requisite fee of $180.00, as well as any fee deficiencies or overpayments to Deposit Account No. 16-1340. A duplicate copy of this page is enclosed herewith.

Respectfully submitted,

Dated: December 8, 2009

Thomas J. Perkowski
Reg. No. 33,134
Attorney for Applicants
Thomas J. Perkowski, Esq., P.C.
22 Thornal Circle
Darien, Connecticut 06820
203-357-1950
http://www.tjpatlaw.com